



# **Use of a Conceptual Model of the Mission Space (CMMS) in Model and Simulation Development: The JWARS/JSIMS CMMS (J<sup>2</sup>CMMS)**

**LTC Terry W. Prosser  
Deputy Director, JWARS  
prossert@paesmtp.pae.osd.mil**

**1997 Department of Defense Senior Technologists Conference**

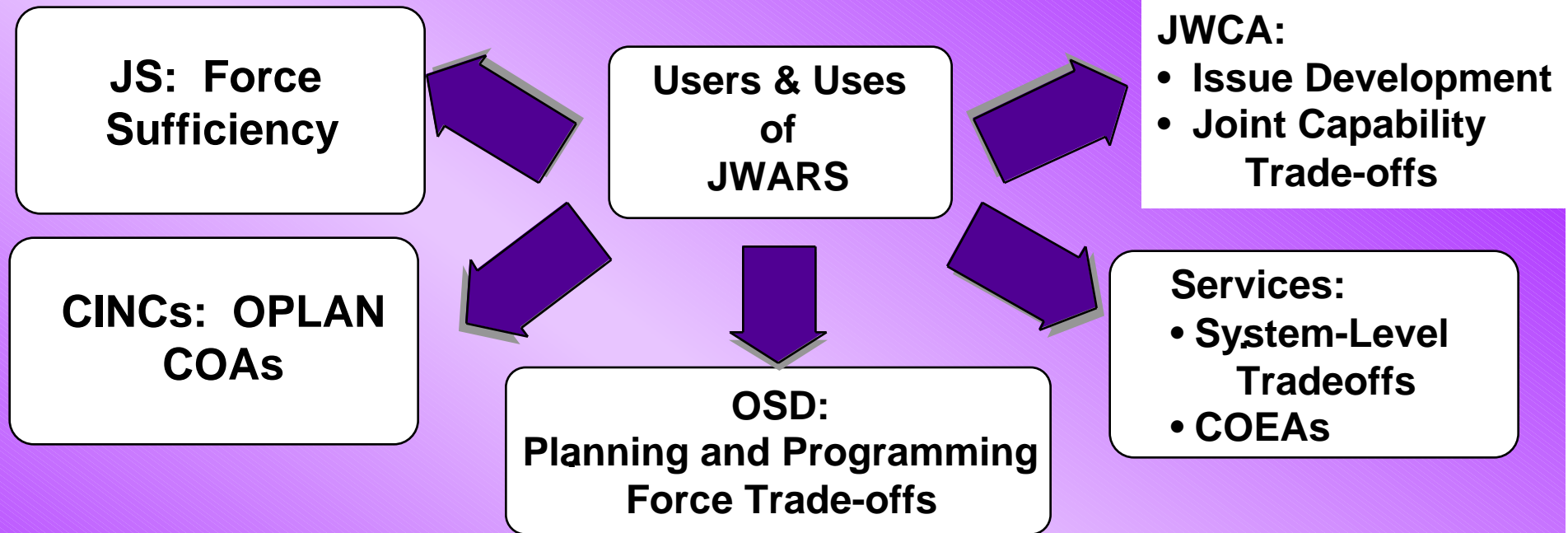
**February 11, 1996**

## **J2CMMS: Linking Through A Common View of the Battlespace**

- Background**
- J2CMMS Purpose**
- Linkage Methodology**
- An Example**
- Lessons Learned**



# JWARS Mission & Analytic Needs



## JWARS Mission

Develop a fully integrated, state-of-the-art, closed-form model of joint warfare. The model must:

- Represent uniquely joint functions and processes and component warfare operations.
- Be based in joint doctrine and capable of representing future warfare.
- Aid in force structure analysis, acquisition analysis, and CINC course of action analysis.

**Principal rationale for developing a common view of the Mission Space include:**

- **Opportunity for reuse given common Mission Space**
- **Opportunity to leverage work between two programs**

**Knowledge Acquisition**

**High-Level & Detailed Design**

**Implementation**

**Knowledge acquisition, or Mission Space Analysis is the area of greatest overlap between JWARS & JSIMS**

# J2CMMS - Purpose

- Conduct knowledge acquisition (KA) of the joint mission space through *research and analysis*
- Document the results in a J2CMMS repository
- Medium to transfer the knowledge to OO analysts to start OO knowledge engineering activities
- Form a basis for VV&A activities

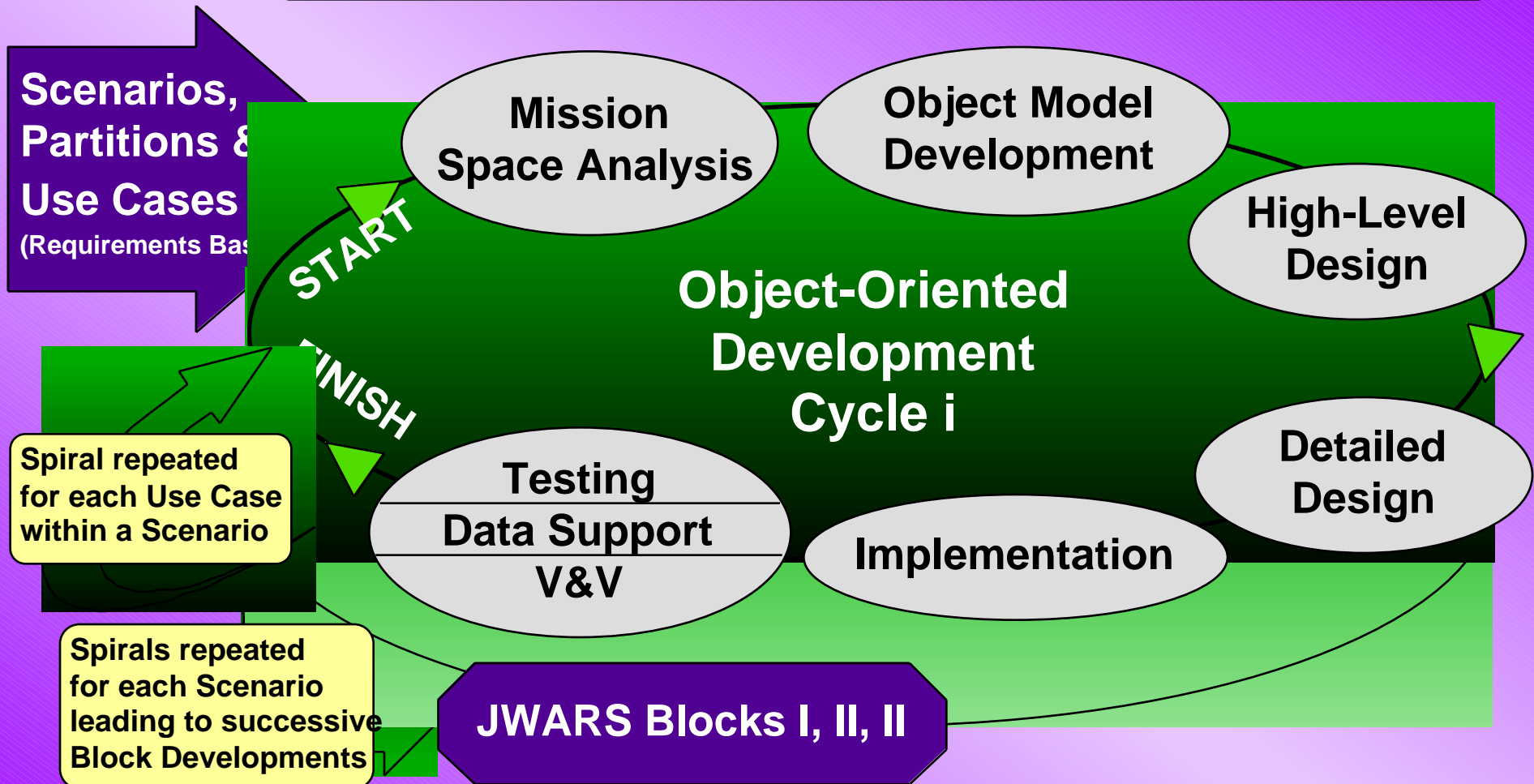


## **J2CMMS Methodology - Principles**

- KA is doctrine based, not SME based**
- Support model development through implementation**
- J2CMMS development is an integral part of the JWARS software engineering process**
- The J2CMMS will evolve as implementation, and therefore additional details, are required**
- Mission space analysts are integrated into the total software engineering process**
- Scenarios provide needed context to the research & analysis**



# J2CMMS and the JWARS Software Development Process

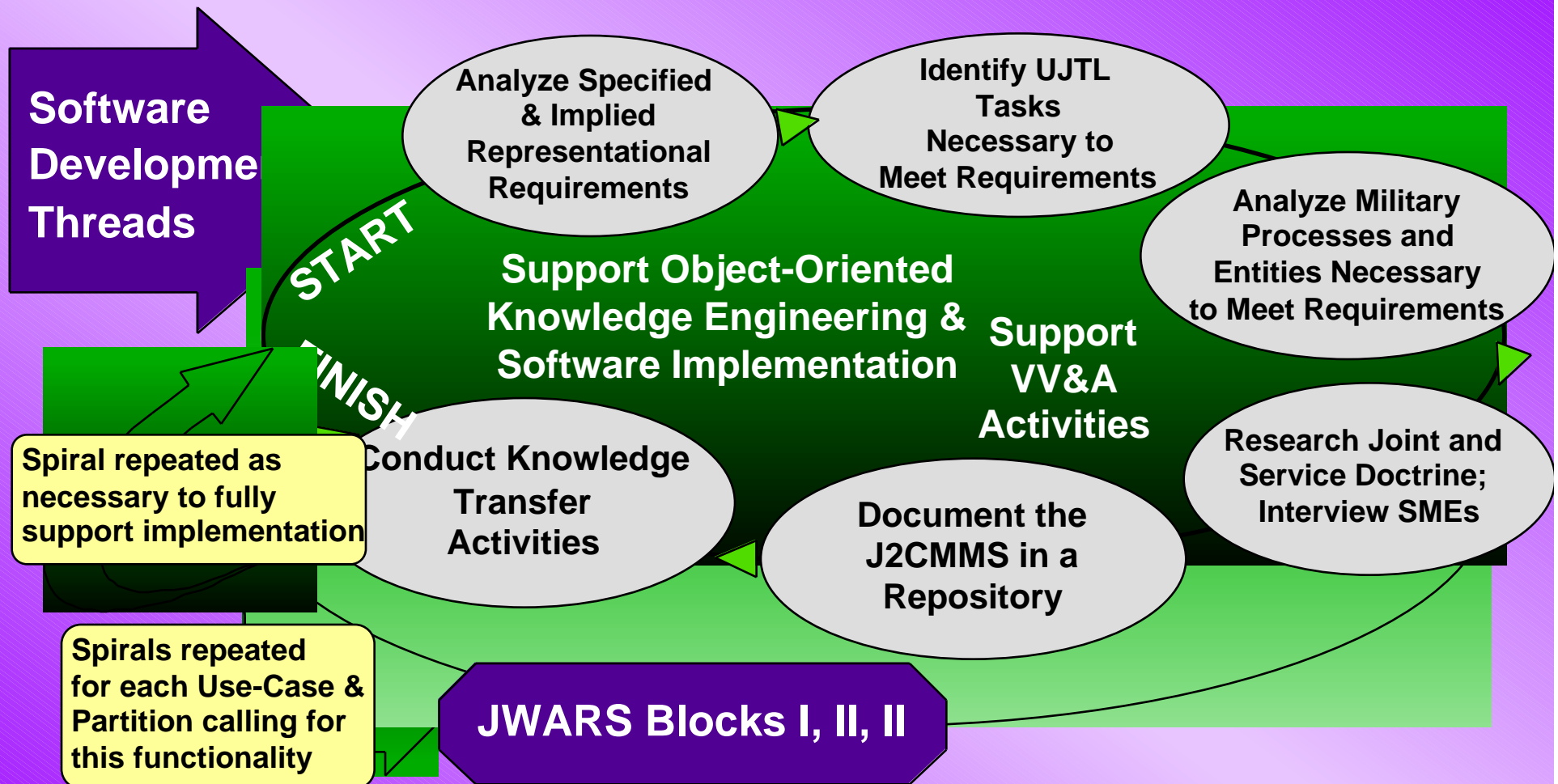


- Spiral approach
- Government guidance and control at every step
- Gov't subject matter experts critical at every step





# JWARS J2CMMS Development Process



- Process adapted to JWARS spiral development approach
- J2CMMS evolves as model development iterates
- Process/entity (functional) product, not object-oriented





## Example - Developing CMMS for *Battlespace Perception*

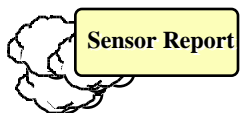
| Entity            | Process  | Action   |
|-------------------|--|--|
| Staff<br>Sections | <ul style="list-style-type: none"> <li>Conduct IPB</li> <li>Predict enemy courses of action</li> </ul>                         | <ul style="list-style-type: none"> <li>Analyze weather</li> <li>Analyze terrain/ocean</li> <li>Analyze enemy</li> </ul>  |
| Staff<br>Sections | <ul style="list-style-type: none"> <li>Plan intelligence collection operations</li> </ul>                                      | <ul style="list-style-type: none"> <li>Identify targets</li> <li>Identify collectables</li> <li>Associate with time &amp; space</li> <li>Match sensors with targets</li> </ul> |
| Sensors           | <ul style="list-style-type: none"> <li>Conduct collection operations</li> </ul>  | <ul style="list-style-type: none"> <li>Prioritize targets</li> <li>Allocate assets</li> <li>Plan missions</li> <li>Execute missions</li> <li>Report results</li> </ul>         |
| Staff<br>Sections | <ul style="list-style-type: none"> <li>Process incoming sensor reports</li> <li>Relate reports to dynamic situation</li> </ul> | <ul style="list-style-type: none"> <li>Receive reports</li> <li>Route reports</li> <li>Correlate information to SITMAP</li> <li>Update knowledge</li> </ul>                    |
| Staff<br>Sections | <ul style="list-style-type: none"> <li>Update assessment of enemy activities</li> </ul>  | <ul style="list-style-type: none"> <li>Match current situation to predicted courses of action</li> <li>Decide which course of action the enemy is following</li> </ul>         |



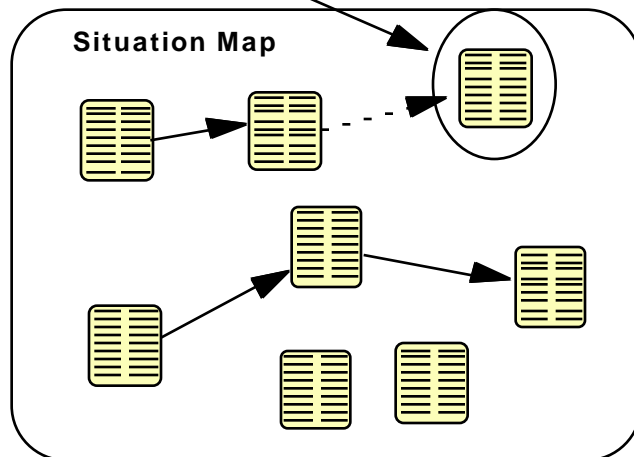
# An Example: Perception and the Fusion Process

## Goal:

- Relate incoming sensor reports to a virtual SITMAP - - Correlation
- Compare perception (SITMAP) to expectations (IPB) - - Assessment

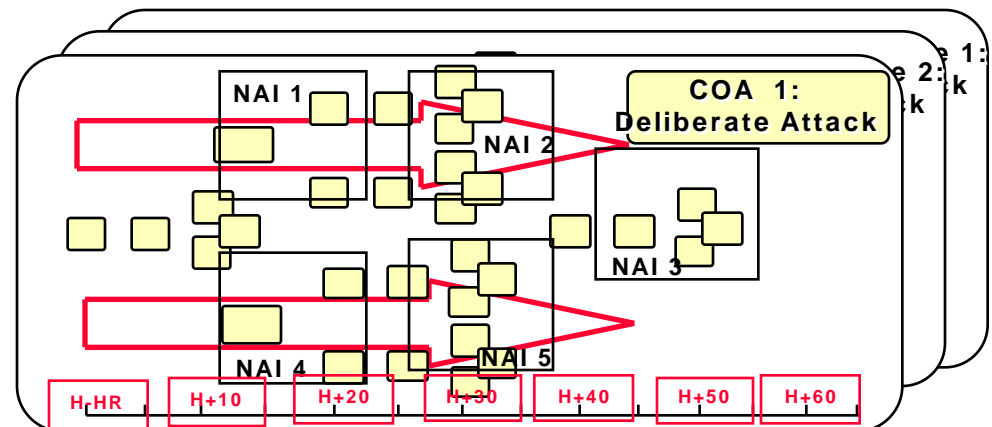


## Correlation



SITMAP contains perceived threat entity matrices.  
Each matrix reflects information reported by sensors.

## Assessment



Threat Course of Action (COA) templates describe expected activities within NAIs over time.

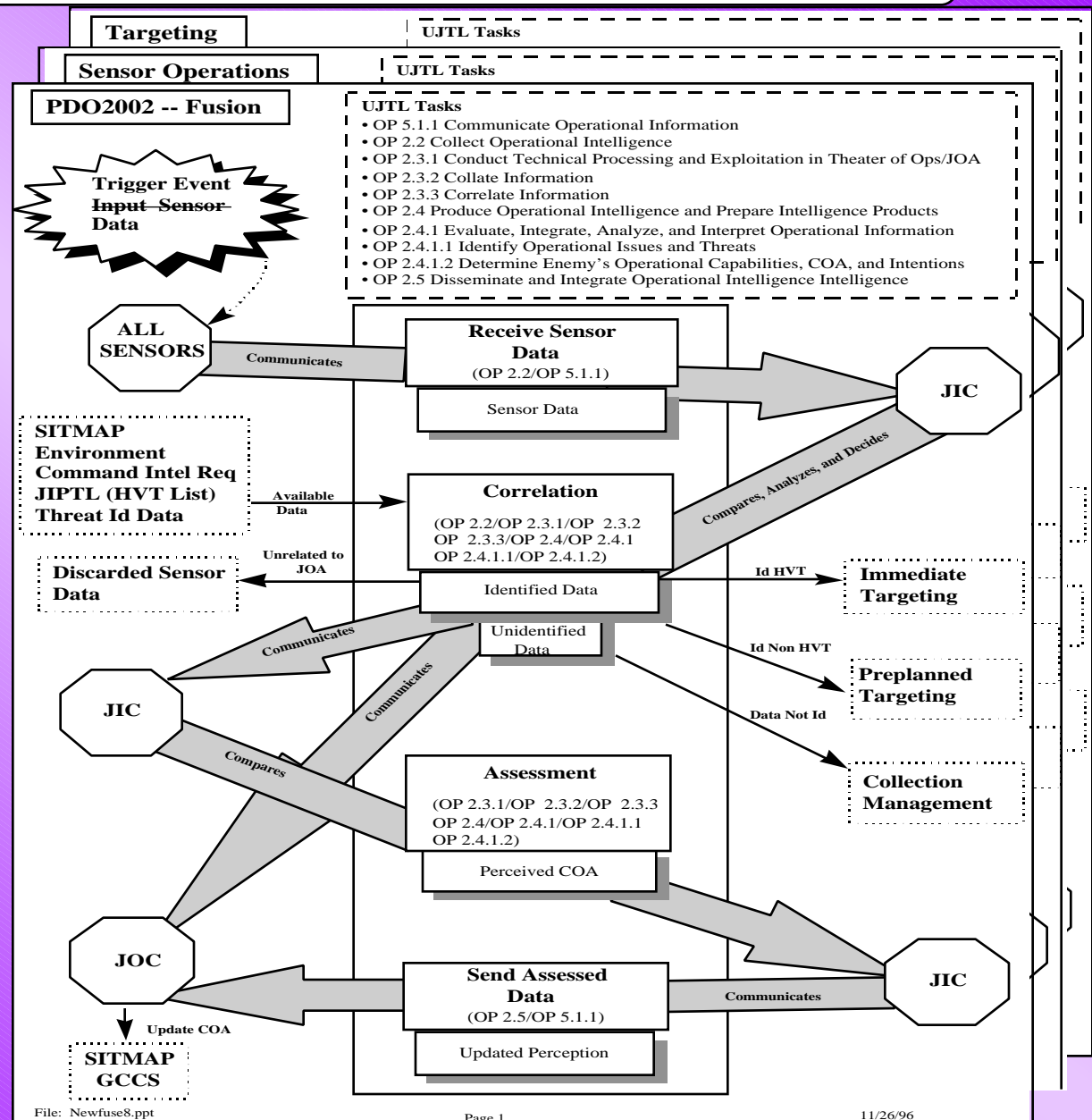
Algorithm(s) find “best fit” between known situation (Situation Map) and Threat templates created as user input during IPB



# J2CMMS Example: Intelligence Fusion Entities & Process

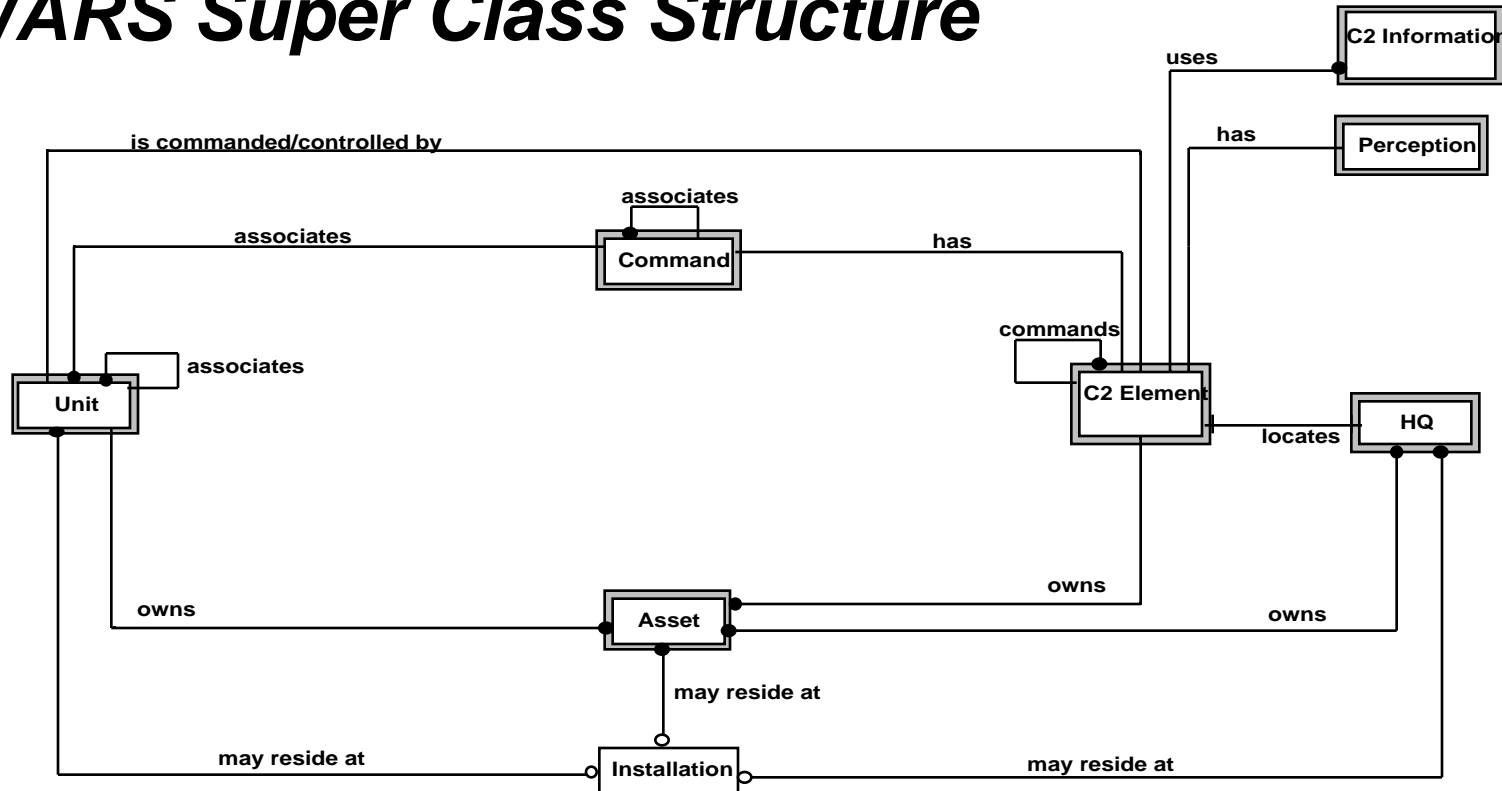
## Desired Outcomes:

- Identify relevant entities
- Identify processes & tasks
- Describe interactions given a common reference scenario
- Describe outputs from the processes and interactions
- Identify pertinent UJTL tasks
- Document references and sources
- Provide a *medium for knowledge transfer* to the object analyst and software engineer



## JWARS OOA Object Model: Object Classes Related to Intelligence Fusion

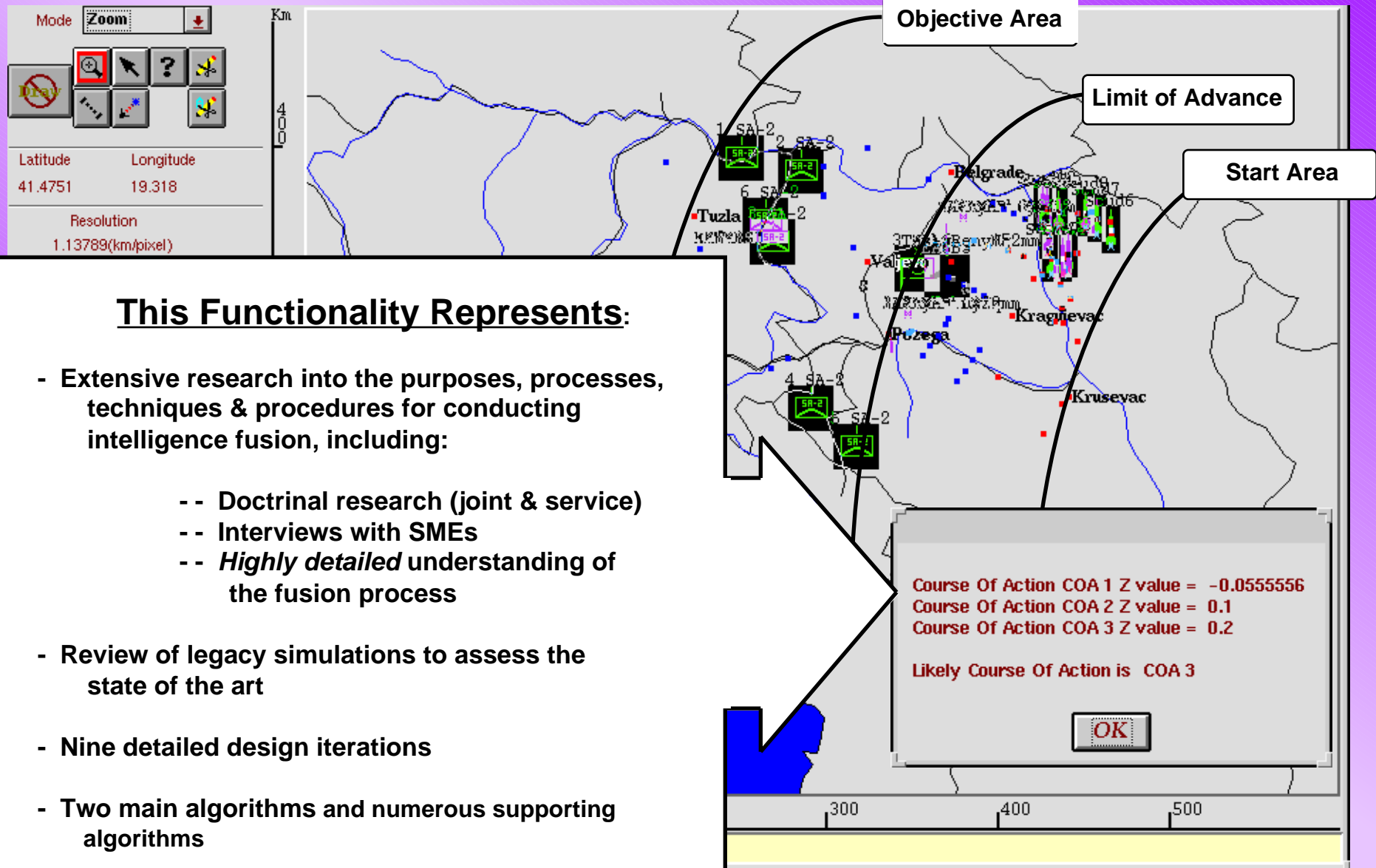
### *JWARS Super Class Structure*



obj\_high\_level  
Thu Jan 2 13:47:48 199  
Class Diagram



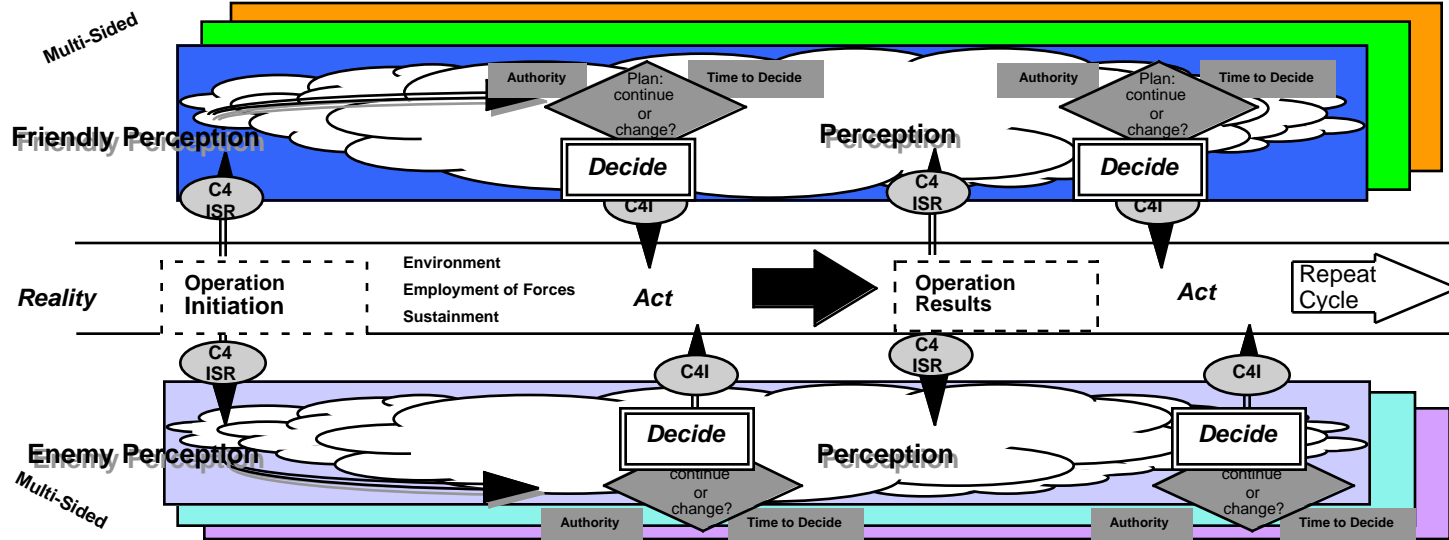
# The Bottom Line: *J2CMMS Must Support Implementation*



***Role of the Mission Space Analysts:***

- Knowledge transfer
  - - Critical to project success
  - - Very time consuming
  - - Necessary during OOA, OOD and implementation
  - - Difficult - different backgrounds required for the KA and KE functions
  - - STATEMATE CASE tool facilitates
- Follow processes through implementation
  - - Monitor functionality evolution
  - - Head team supporting software engineer

# J2CMMS Lessons Learned



***Know your core reasons for building the model:***

- These influence KA, design & implementation;

**Some JWARS examples:**

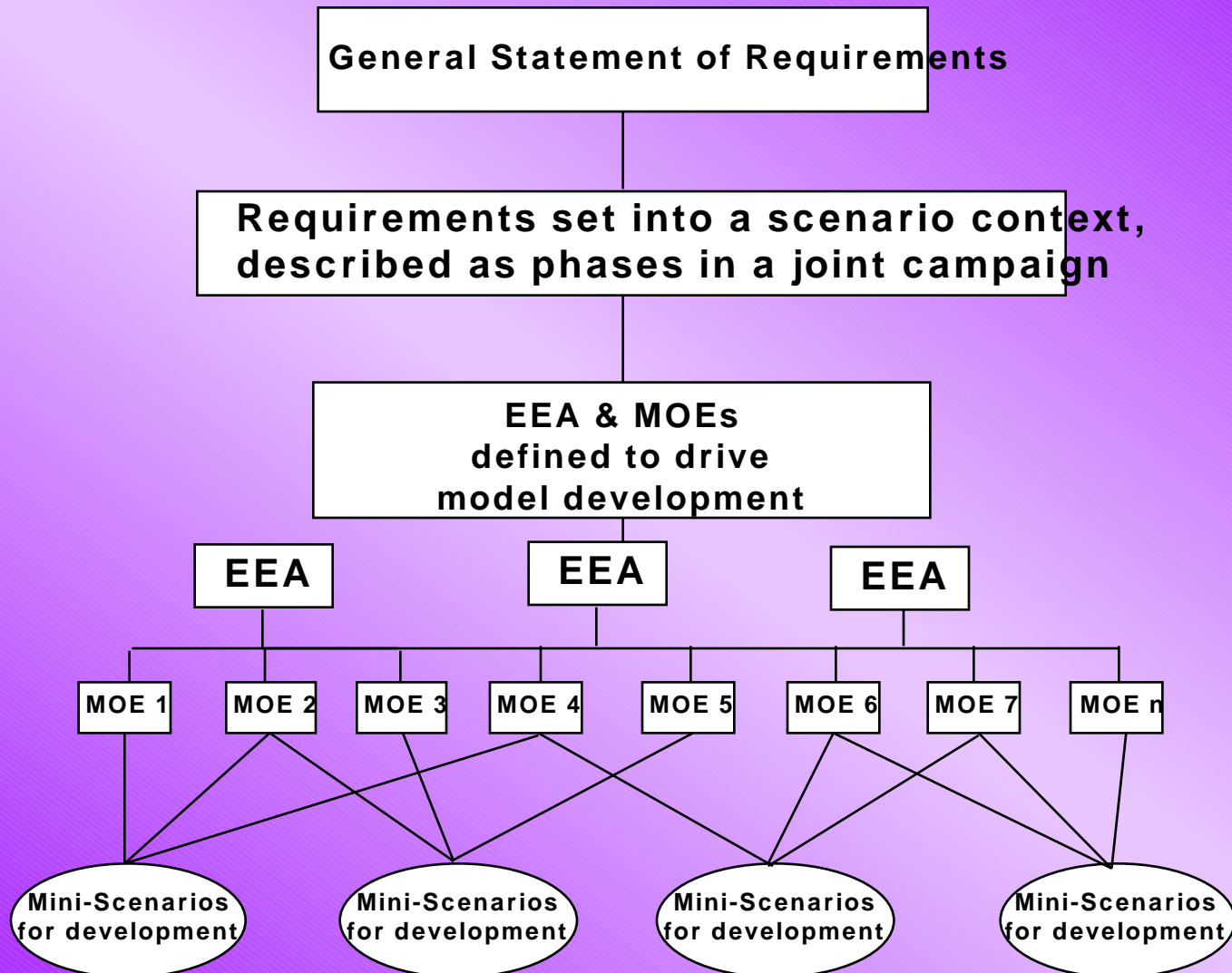
- - Effect of information on battle outcomes
- - Maintain balance between services
- - Emphasize uniquely joint functions
- - Provide timely & transparent results



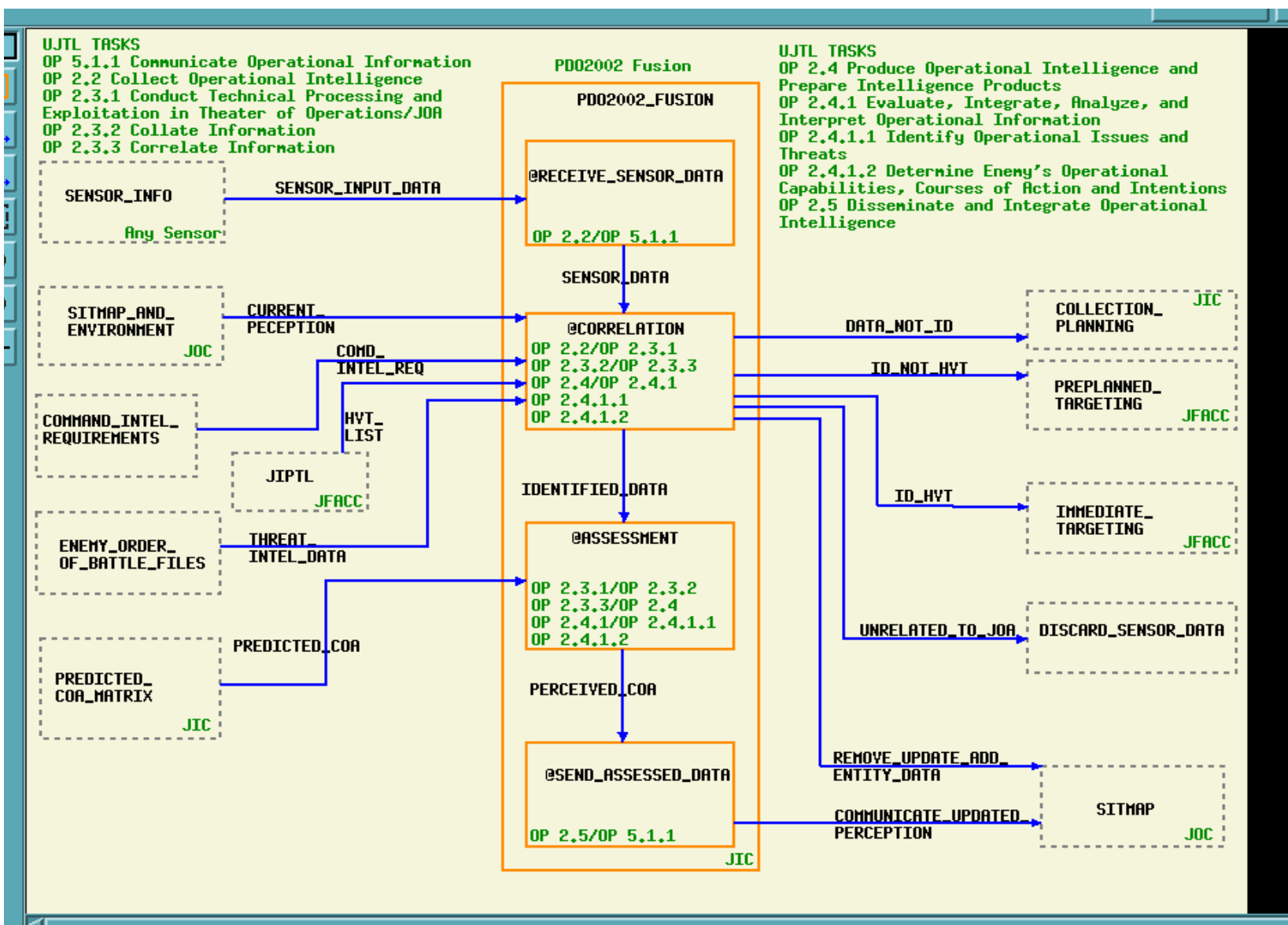


# Back-ups

# Software Development Process: Requirements Definition



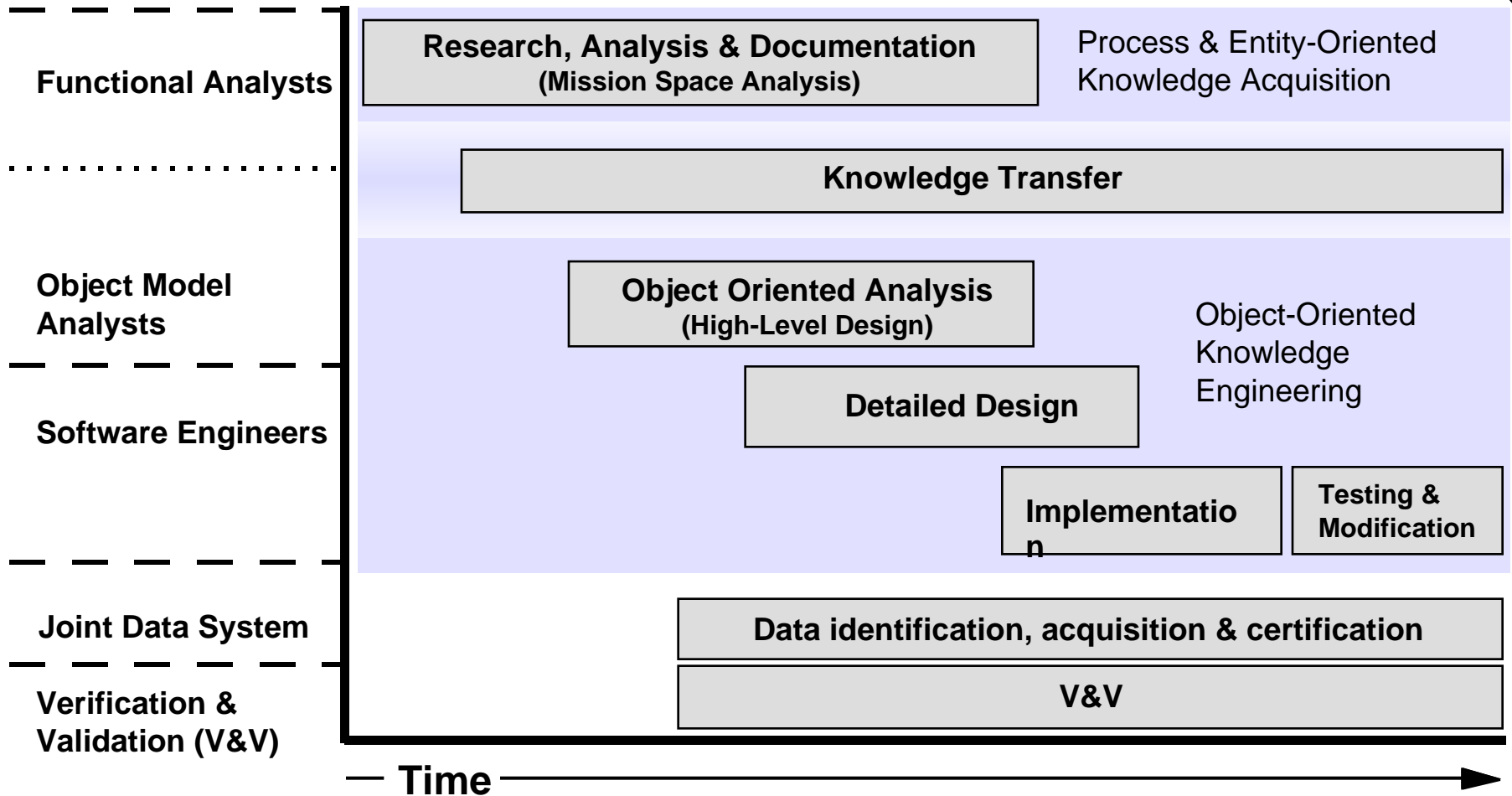
## Mission Space Model STATEMATE Output: Fusion Activity Diagram





# JWARS Software Development Process

## Thread-Based Steps in Iterative Development



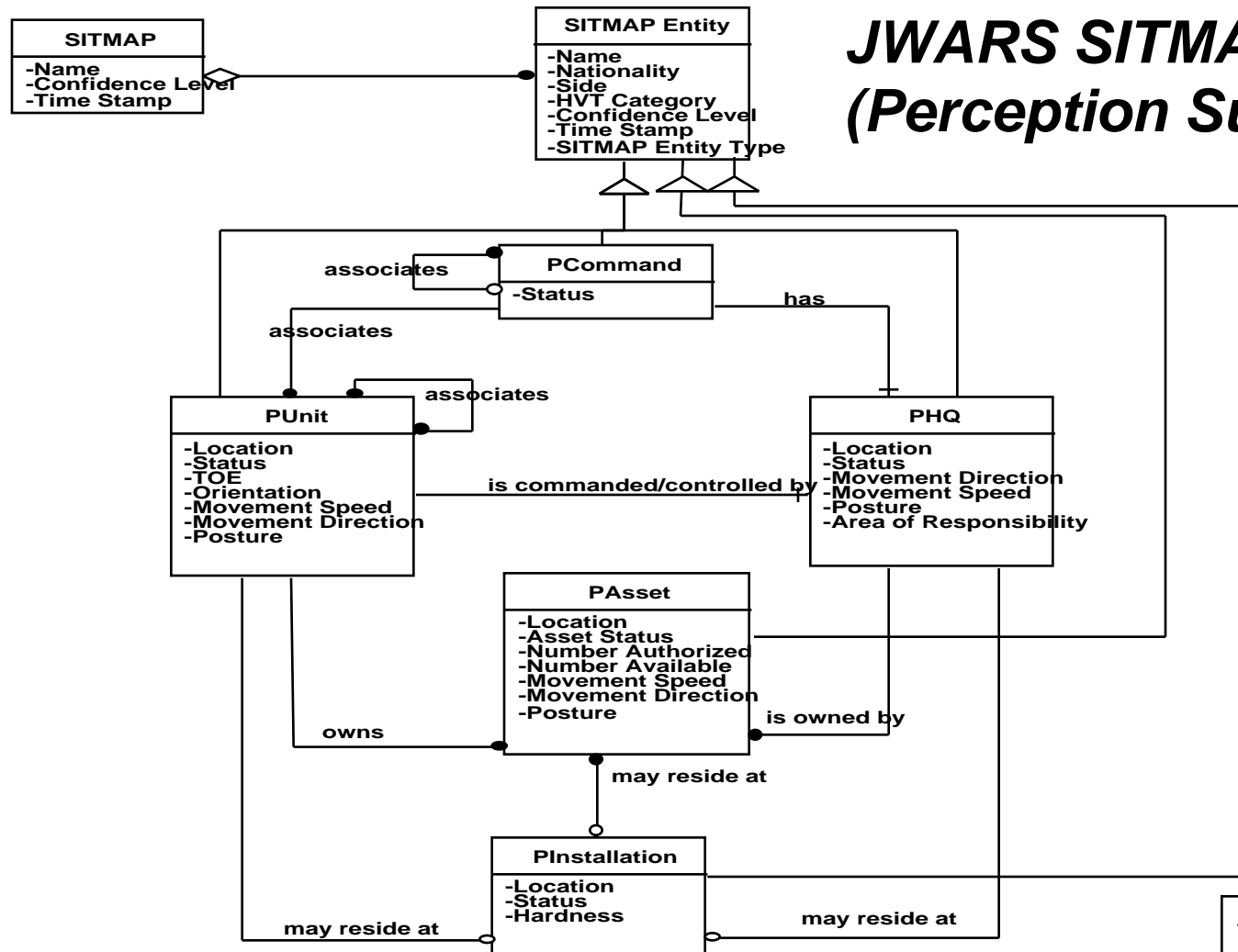
Steps are completed for each software development *Thread* that are part of each JWARS *Use Case*

OOA: Object-oriented (OO) analysis; in OMT terms, the first OO abstraction of the mission space.

OOD: OO design; in OMT terms, the final detailed object design & integration with the system architecture.

## JWARS OOA Object Model: Object Classes Related to Intelligence Fusion

### JWARS SITMAP Classes (Perception Sub-Class)



obj\_sitemap  
Thu Jan 2 13:48:57 199  
Class Diagram

## JWARS OOA Object Model: Object Classes Related to Intelligence Fusion

### JWARS IPB Classes (Perception Sub-Class)

